

An Interactive Monitoring Robot for Dementia Mitigation via Daily Conversations with Multiple LLMs

Masayuki Numao¹, Masahiro Kawamura²

¹Tokyo Information Design Professional University

²University of Electro-Communications

numao@tid.ac.jp, k2231032@edu.cc.uec.ac.jp

Abstract

This study presents the development of an interactive cognitive function assessment system designed for everyday use, to address the growing number of dementia cases and the rising burden of care in an aging society with a declining birth rate. The proposed system integrates multiple large language models (LLMs) to achieve this goal.

The system consists of three distinct LLM modules: (1) Detector, responsible for identifying user intentionality, (2) ChitChat, facilitating casual conversation, and (3) StrictTask, managing task-oriented dialogues. These modules are orchestrated using the LangChain architecture to function as a cohesive assessment system.

Evaluation experiments were conducted to assess the accuracy of the HDS-R (Hasegawa's Dementia Scale-Revised) in the proposed LLM-based system and compare it with existing rule-based systems. The evaluation employed pre-designed conversational scenarios as test cases, identifying specific errors in each scenario.

Results indicated that the proposed system demonstrated flexible handling of user interruptions and adaptable responses to user inputs, while also probabilistically managing the strict conditional branches dictated by HDS-R. In comparative evaluations, the proposed system exhibited lower accuracy in strict conditional branching compared to the existing rule-based systems. However, it significantly outperformed the conventional systems in terms of dialogue quality, offering features such as flexible responses, immediate reactions, and engaging chit-chat. These features improve user engagement and overall system usability, demonstrating the advantages of the proposed system as a user-friendly cognitive assessment tool.

Introduction

With the advancement of AI technologies, such as large-scale language models (LLMs), an increasing number of el-

derly individuals are expected to utilize cutting-edge technologies to support their health needs, including self-care management and health promotion activities. Previous studies indicate that voice-activated chatbots are highly beneficial for elderly individuals in health-related communication and information retrieval (Brewer, 2018). Additionally, virtual assistants are being employed for medication reminders (Dahlke 2017) and to support elderly cancer patients undergoing home therapy (Piau 2019).

Caregivers of individuals with dementia often experience depression, stress, and declining health due to a lack of knowledge and effective coping strategies for managing complex dementia care. This can negatively affect the quality of care provided to people with dementia. AI has demonstrated the potential to alleviate these challenges by both reducing the burden on caregivers through the direct provision of elderly care and offering education and emotional support to caregivers.

This study aims to develop a dialogue system integrating chat-based and questionnaire-driven dialogues to facilitate the interactive assessment of cognitive function using HDS-R. The aim is to create a dialogue system that can be seamlessly incorporated into everyday life.

We first listed up the system requirements for 2 main functions: Daily continuous monitoring and HDS-R cognitive function assessment. These requirements emphasize flexibility, user engagement, and adaptability to ensure effective interaction with elderly individuals.

The Large Language Model (LLM), which has recently garnered significant attention, possesses the capability to facilitate not only question-answering systems but also chatting and diagnostic application systems. This serves as the foun-

dation of this research. Specifically, by utilizing prompt engineering to direct LLMs, we will design prompts that encompass a chat function, a task-oriented dialogue function, and an intention recognition function, ultimately combining them into a unified system.

Related Works

Irfan et al (Irfan 2023) provided hyper-parameter conditions for applying LLM to a companion robot for chatting with elderly people. They added a persona to the robot that proactively asked about emotions and health, and talked about itself, based on feedback from the elderly. Experimental results showed that persona prompts can be modified to use more follow-up questions and to give the robot a deeper personality (e.g. pre-defined likes, dislikes, memories), thereby improving the believability of the character.

Suzaki et al (Suzaki 2023) created a scenario for running HDS-R on a scenario-based dialogue system, and demonstrated the possibility of diagnosis using a dialogue system. By adding the function of meta-dialogue control, their system enabled flexible transitions according to the user's intention, in addition to controlling the order of complex scenarios, thus fulfilling the implementation requirements of HDS-R.

Requirements

Requirements for Daily Constant Monitoring

1. Integration of Task-Oriented Dialogue and Chit-Chat

For effective daily monitoring, it is crucial to incorporate both task-oriented dialogue, such as schedule checks, diagnostic conversations, and story reading, along with casual everyday conversations (chit-chat). The system must seamlessly transition between task-based interactions and free-flowing conversations depending on the context and user needs.

2. Robot-Initiated Dialogue

Unlike most chatbot systems that depend on user initiation, this system requires the robot to initiate conversations to maintain engagement with elderly users. This helps maintain continuous dialogue and ensures that users remain engaged.

3. Recognition of Activities of Daily Living (ADL) Recognizing the user's daily activities allows the system to provide more contextually appropriate responses. This requires a multimodal dialogue function capable of integrating visual information, enhancing the system's understanding of the user's situation.

Requirements for HDS-R Assessment

The HDS-R (Hasegawa's Dementia Scale-Revised) includes nine questions covering the patient's age, date, location, and other relevant details. Scoring is defined for each question. The total score is 30 and if it is less than 21, patients may have dementia. Each question has a specific requirements concerning the order, timing, and delivery of the questions. The successful integration of HDS-R into a dialogue system necessitates addressing the following requirements:

1. Sequential Control

Sequential control involves managing conditional branching and repetition. For example, in Question 2, the system must repeatedly prompt the user until they correctly state all dates and days of the week. In Questions 3 and 7, the system should provide hints based on the user's responses. Questions 5 and 6 require branching, as subsequent questions depend on whether the user's answer is correct. In Question 9, where the user is asked to list vegetables, the system must be capable of asking follow-up questions like, "Do you know any other vegetables?"

2. Provision of Hints

The system should offer hints when users say, "I don't know," request help, or remain silent. Hints must align with the context of each question. Specifically, for Questions 3 and 7, the system must have pre-defined question-hint pairs to ensure that only appropriate hints are provided.

3. Meta-Level Dialogue

Meta-level dialogue, as proposed by Suzaki (Suzaki 2023), enables users to switch between topics through specific commands. This feature allows users to pause, skip questions, or ask the robot to repeat questions. Since elderly users may struggle with hearing or comprehension, implementing meta-level dialogue is crucial to ensure a smooth user experience.

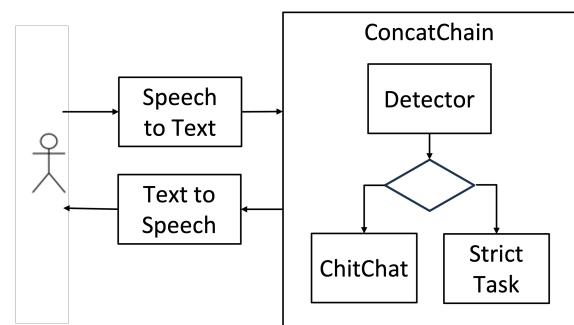


Figure 1. Multiple LLM Structure

Implementation by Multiple LLMs

The structure of the system is depicted in Fig. 1. The system is composed of four key components: Detector, ChitChat, StrictTask, and ConcatChain. Each is a chain model incorporating Large Language Models (LLMs), with outputs adjustable through hyperparameter tuning. These chains are part of the Langchain library, a framework provided by OpenAI for integrating LLMs with prompts. ConcatChain uses Langchain to combine multiple models. Speech2Text and Text2Speech functionalities are facilitated by Vosk and VoiceVox, respectively.

System Flow The system processes inputs such as user speech, periods of silence, and image data. Based on this information, the Detector module assesses user intention. Depending on the detected intent, the system activates either ChitChat or StrictTask, checking if the intent is part of a predefined task list. The selected model's output is processed by ConcatChain and delivered as speech to the user.

Detector LLM

Detector is responsible for determining user intent. A few-shot prompting approach is used, incorporating multiple examples to enhance response accuracy. Chain-of-Thought (CoT) prompting can enhance accuracy, but passing the entire dialogue history often causes delays, impairing conversation fluidity. Thus, the few-shot approach is optimized by increasing the variety of example patterns and estimating

Prompt for Detector LLM

The AI estimates the next topic based on pairs of the most recent topic and conversation history. Additionally, if the most recent topic is HDS-R, HDS-R will continue to be the next topic until the user requests a pause.

- Examples of topic estimation: Greeting: "Dinner was delicious today." → Meal
- HDS-R: "I'm 24 years old." → HDS-R
- HDS-R: "At home." → HDS-R
- Weather: "I haven't been feeling well lately." → Health
- Hobby: "Listening to music makes me happy." → Hobby
- Hobby: "I don't know." → Hobby
- HDS-R: "I'm done." → End
- Greeting: "Good morning." → Greeting
- HDS-R: "I want to pause." → Pause
- Greeting: "Please diagnose me." → HDS-R

Following the examples above, please estimate the next topic from the pair of the most recent topic and conversation history below.

Prompt for ChitChat LLM

Please take a deep breath.

As a care staff member, your job is to listen to the client.

Please show empathy and offer kind, considerate words to the client.

Answer questions clearly and concisely in no more than two sentences.

Whenever possible, ask questions to learn more about the client's interests.

If you don't know the answer to a question, honestly say, "I don't know."

Additionally, refer to the context information below when necessary to support your responses:

####

Facility Usage Q&A: {qa_content}

####

user intention based on both current utterances and previous topics. The actual prompt is shown below.

ChitChat LLM

ChitChat generates casual dialogue. It retains the past four dialogue exchanges (WINDOW SIZE=4) and operates with a temperature parameter of 0.4 to balance response variety and consistency. Given that elderly users are the target group, the prompts are tailored to daily life topics. Additionally, drawing from prior research (Brewer 2018), the system proactively introduces topics to sustain user engagement. To mitigate LLM hallucinations (fabricating plausible but false information), the system is instructed to respond with "I don't know" when faced with unanswerable questions. The actual prompt is shown below.

StrictTask LLM

StrictTask handles task-oriented dialogues, with a temperature setting of 0 for deterministic outputs. Since this study focuses on interactive cognitive function assessment, particularly HDS-R (Hasegawa's Dementia Scale-Revised), maintaining dialogue state is crucial. As LLMs alone cannot track dialogue states, Langchain's dialogue memory capabilities are employed to sustain context.

The prompts for StrictTask reference prior works (Kojima 2023) and emphasize slot-filling techniques, instructing the LLM to collect user information step-by-step. Sequential control is guided by research (Zhang 2023), aligning the dialogue structure with predefined slots. Prompts also specify the provision of hints and meta-dialogues to ensure smooth transitions and support during the HDS-R assessment.

Challenges specific to HDS-R include selecting appropriate words for generated questions and synchronizing visual outputs with speech. Questions involving calculations or memory recall require validating user responses as correct or incorrect. Moreover, HDS-R image-based questions pose limitations since LLMs lack fine control over synchronized image and speech outputs. The actual prompt is shown below.

Prompt for StrictTask LLM

The AI will fill in the user slots step-by-step based on the following conditions:

- When the user's utterance includes a request such as "Diagnose me" or "Perform dementia diagnosis," the questioning process will begin.
- The AI will ask the user questions one at a time in a conversational format (referring to the dialogue history).
- Based on the responses obtained from the questions, the AI will fill in the user slots sequentially, one by one.
- The AI will populate the user slots using the information obtained through the conversation.
- If the user's response involves asking for repetition or skipping a question, the AI will transition accordingly.
- If the user says they don't know, the AI will either provide a hint or move on to the next question.

Prerequisites:

- Slot 6 will only be filled if Slot 5 is answered correctly.
- Slot 8 will only be filled if Slot 7 is answered correctly.

User Slots:

1. Age
 2. Today's year, month, day, day of the week
 3. User's current location
 4. Repetition of three words (sakura, cat, train)
 5. Calculation: 100 minus 7
 6. Calculation: 93 minus 7
 7. Digit span backward (2, 8, 6)
 8. Digit span backward (3, 5, 2, 9)
 9. Recall of the three words repeated earlier
 10. Naming as many vegetables as possible
-

Algorithm 1: Generate input => output

```

1: text = input
2: if text == None then
3:   text = "Please ask a question because the user is remaining silent"
4: endif
5: preintent = intent
6: intent = Detector(text, preintent)
7: if intent in StrictTask.list then
8:   output = StringTask(text, intent)
9: else if preintent in StrictTask.list and
  (intent == "Break" or intent == "Finish") then
10:  output = StrictTask(text,intent)
11:  Init StrictTask
12:  Init ChitChat
13: else
14:  output=ChitChat(text)
15: end if
16: return output

```

ConcatChain

ConcatChain orchestrates the interaction between Detector, ChitChat, and StrictTask using Langchain. Its logic is described in Algorithm 1. Initially, user input and the previous topic (pre-intent) are fed into Detector to infer the current intent. If the intent corresponds to a task in the StrictTask scenario list, StrictTask is activated; otherwise, ChitChat is invoked.

If the pre-intent relates to a task and the current intent signals task completion, both ChitChat and StrictTask histories are reset. This ensures accurate dialogue flow, preventing ChitChat responses from disrupting task-oriented interactions. Finally, the model output is returned as the system's response.

Evaluation Experiments

To validate the system, we conducted three evaluation experiments: an evaluation of stable dialog management, accuracy of HDS-R scoring, and compared with an existing system, and a demonstration in a nursing home.

Evaluation of Stable Dialog Management Experimental Setup

This evaluation assesses the system's ability to maintain stable dialogue flows, even when responses are not directly related to the questions. 12 assumed scenarios were created, with several dialogue patterns tested for each to evaluate that the system maintain the stable dialog. Success was defined as a dialogue proceeding as intended based on the scenario.

Failures were documented with examples. The assumed scenarios are listed as follows:

1. All questions answered correctly.
2. User provides year, month, and day in separate parts.
3. Calculation errors and conditional branching evaluation.
4. Recapitulation errors and conditional branching evaluation.
5. User requests hints.
6. Ambiguous responses (e.g., filler words, uncertainty).
7. User requests repetition.
8. Naming vegetables and confirming the number.
9. User probes for the answer.
10. User provides incorrect answers to all questions.
11. Partial completion of questions.
12. User requests interruption.

Results and Discussion

Example dialogues demonstrated that individual mistakes did not compromise overall dialogue flow. Success was consistent in scenarios involving correct answers or date variations. However, errors occurred in scenarios involving calculation mistakes or recapitulation failures. LLMs often produced plausible but incorrect answers due to a lack of certainty regarding correctness.

When ambiguous responses like "93 or 92?" were given, the system treated them as incorrect without seeking clarification—appropriate behavior from the LLM but problematic for accurate diagnosis. Despite occasional errors, the diagnostic outcome was generally unaffected if the dialogue was otherwise ideal.

Calculation or recapitulation errors resulted in additional questioning but did not interfere with the overall dialogue flow. Hint and repetition requests were functional, though the system sometimes provided direct answers instead of hints. While prompt adjustments were attempted, controlling this behavior proved challenging, suggesting the need for alternative methods.

The system appropriately managed filler words and irrelevant responses in the vegetable-naming task. However, cases where users began answering with the day of the week caused inconsistencies, such as skipping the year or only requesting the month and day. Dialogue history was not always fully leveraged, indicating that LLM fine-tuning was light and not comprehensive.

Questions probing for answers often led to LLMs revealing the correct response. In such cases, human supervisors were required to review the dialogue history to assess diagnostic validity.

Comparative Evaluation

The proposed system was compared to an existing system developed by Suzaki et al. (Suzaki 2023). Key evaluation

criteria included diagnostic functionality, chit-chat capability, and dialogue fluency (response processing speed). The results showed that the proposed system effectively managed user interruptions and adapted well to varying user

Dialog Example

User: Diagnose.

Bot: Yes, we will start the diagnosis. First of all, please tell us your name.

USER: My name is Yamanaka.

Bot: Yamanaka, nice to meet you. So, first of all, please tell us your age.

USER: I am 24 years old.

Bot: Thank you very much. Next, please tell us today's date and day of the week.

User: Thursday, 12 October 2023

Bot: Thursday 12 October 2023. Thank you very much. Next, please tell us your current location.

USER: At home.

Bot: Home, yes. Thank you very much. Next, please tell me if you remember the three words I said. Cherry blossom, cat and train.

User: Cherry blossom cat train.

Bot: Thank you very much. Next, I would like to try a simple subtraction. First, 100 subtract 7, how many?

User: 93

Bot: Then tell me the value of 7 subtracted again.

User: 86

Bot: Thank you very much. Next, tell me the numbers I said backwards: 2, 8, 6.

User: 682

Bot: Thank you. Next, please reverse the numbers I gave you: 3, 5, 2, 9.

User: 9253

Bot: Thank you very much. Next, could you tell me the three words you just learned?

User: Cherry cat train.

Bot: Thank you very much.

Bot: Finally, please answer as many vegetables as you know. Go ahead.

User: cabbage carrot Chinese cabbage onion green pepper

Bot: Thank you very much. Are there any other vegetables you know?

User: tomato burdock burdock taro yam sweet potato spinach

Bot: Thank you. Are there any other vegetables you know of?

User: bok choy shiso shiso

Bot: Thank you. Are there any other vegetables you know of?

User: No more.

Bot: Understood. This concludes the question. Thank you for your time.

Comparison items	Proposed system	Existing system
Answer all questions correctly	⊙	⊙
Saying date incorrectly	△	○
Calculation error	△	○
Recitation error	△	○
Request for hints	○	○
Vague response	×	×
Requests for repetition	○	○
How many vegetables	○	×
Questioning for answers	×	○
Incorrect answers	△	○
Partial questions	○	×
Chatting	⊙	×
Dialogue fluency	⊙	×

Table 1. Comparison Table

inputs. The existing system also handled year, month, day, and weekday inputs discretely, reducing user stress during the HDS-R assessment.

However, the proposed system demonstrated higher flexibility, especially in the vegetable-naming task. This flexibility stemmed from the stochastic nature of LLM response generation, which, while advantageous in free-form dialogue, proved less reliable in strictly controlled diagnostic contexts.

The existing system's rule-based approach ensured consistent performance by predefining response patterns, whereas the proposed system's probabilistic nature sometimes led to deviations from expected behavior. This highlights the need for further refinement in the proposed system's conditional branching and control mechanisms.

Future Issues

The scenario-based accuracy evaluation revealed that the current system has not yet met all required criteria. This study primarily emphasized enhancing the convenience and usability of the dialogue system, as it was designed for elderly users. The goal was to develop a system that is easier for elderly individuals to use compared to existing systems. However, to advance the system to the next stage, it is necessary to ensure that it aligns more closely with the core diagnostic requirements. Abrupt test initiation or poorly worded prompts may cause user anxiety, potentially affecting their test performance.

Therefore, future research should focus not only on usability but also on refining aspects such as pre-test explanations, post-test follow-up, and the choice of wording and voice tone to minimize user tension and anxiety. Evaluating and optimizing these factors will be crucial for improving both diagnostic accuracy and user experience.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number JP22H00547 " Support system for early detection of symptoms and prevention of progression of dementia based on continuous biological and behavioural data."

References

- Brewer R., N., et. al 2018: Accessible voice interfaces. ACM Conference on Computer Supported Cooperative Work and Social Computing, CSCW '18 Companion, pp. 441–446, 2018.
- Dahlke, D. and Ory, M. G., 2017 D.: Emerging opportunities and challenges in optimal aging with virtual personal assistants. Public Policy & Aging Report, Vol. 27, No. 2, pp. 68–73, 07 2017.
- Piau, R. et al. 2019: A smartphone chatbot application to optimize monitoring of older patients with cancer. International Journal of Medical Informatics, Vol. 128, pp. 18–23, 2019
- Irfan, S-M., Naki, K. and Skantze, G. 2023: Between reality and delusion: Challenges of applying large language models to companion robots for open-domain dialogues with older adults, Research Square, 05 May 2023.
- Suzaki, M. and Numao, M. 2023: State Machine-Based Multimodal Dialogue System for the Elderly Care Service, AAI Spring Symposia 2023.
- Wei X. et. al 2023: Chain-of-thought prompting elicits reasoning in large language models, 2023.
- Kojima, S. et. al. 2023: Large language models are zero-shot reasoners, 2023.
- Zhang, X. et. al. 2023: Building task bots effortlessly via schema-guided llm prompting, 2023.